

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended): A computerized method for translating source code into object code, comprising:
 - recognizing a history operator and a history operand in the source code;
 - generating first object code that, when executed, saves a data history associated with an instance of the history operand; and
 - generating second object code that, when executed, performs the history operator on the data history,
wherein the history operator is capable of being referenced directly by the source code.
2. (Original): The method of claim 1, wherein the first object code further saves values assigned to a variable in the data history when the object code is executed.
3. (Original): The method of claim 1, wherein the history operand further comprises an expression of variables and wherein the first object code further saves a result of the expression in the data history.
4. (Original): The method of claim 1, wherein generating first object code further comprises allocating storage for the data history.
5. (Original): The method of claim 2, wherein the data history further comprises program locations where the assignments occurred and timestamps indicating when the assignment was made.
6. (Original): The method of claim 1, wherein performing the history operator on the data history further comprises:

querying the data history based on contents of the data history.

7. (Original): The method of claim 1, wherein the history operand comprises a function and the data history comprises values returned by the function.

8. (Original): The method of claim 7, wherein the data history further comprises program locations where the values were returned and timestamps indicating when the values were returned.

9. (Original): The method of claim 1, wherein the history operand comprises a label associated with a source code statement, and wherein performing the history operator on the data history further comprises:

counting a number of times the source code statement associated with the label was executed.

10. (Original): The method of claim 9, wherein the label is programmer-defined.

11. (Original): The method of claim 9, wherein the label comprises a programming language control construct.

12. (Original): The method of claim 1, wherein performing the history operator on the data history is a function selected from a group consisting of:

summing the data history, averaging the data history, determining a maximum of the data history, selecting an element of the data history, determining a minimum of the data history, determining a number of values in the data history, determining a first entry in the data history, determining a last entry in the data history, determining a subsequence of the data history, performing a reduction operation, and performing a statistical function.

13. (Original): The method of claim 1, wherein:

the history operand comprises a programming language keyword representing a loop; and

the history operator comprises an iteration count of the loop.

14. (Original): The method of claim 1, wherein saving the data history further comprises:
saving the data history in an array, wherein each element of the array comprises a value associated with the history operand at a particular time.
15. (Original): The method of claim 1, wherein saving the data history further comprises;
saving the data history in a linked list.
16. (Original): The method of claim 1, wherein saving the data history further comprises;
saving the data history in a file.
17. (Original): The method of claim 1, wherein performing the history operator on the data history further comprises:
resetting the data history to null.
18. (Original): The method of claim 1, wherein saving the data history and performing the history operator further comprise updating an accumulator.
19. (Currently Amended): A computer-readable medium containing source code, wherein the source code comprises:
a history operand to direct a translator to generate first object code that, when executed, saves a data history associated with an instance of the history operand; and
a history operator to direct the translator to generate second object code that, when executed, performs the history operator on the data history,
wherein the history operator is capable of being referenced directly by the source code.

20. (Original): The computer-readable medium of claim 19, wherein the history operand comprises a variable and the data history comprises values assigned to the variable.
21. (Original): The computer-readable medium of claim 19, wherein the history operand comprises an expression of variables and the data history comprises a result of the expression.
22. (Original): The computer-readable medium of claim 19, wherein the history operand comprises a heap-allocated object.
23. (Original): The computer-readable medium of claim 19, wherein the history operand comprises a function and the data history comprises values returned by the function.
24. (Original): The computer-readable medium of claim 23, wherein the data history further comprises program locations where the values were returned and timestamps indicating when the values were returned.
25. (Original): The computer-readable medium of claim 19, wherein the history operand comprises a label associated with a source code statement, and wherein performing the history operator on the data history further comprises:
counting a number of times the source code statement associated with the label has been executed.
26. (Currently amended): A computer-readable medium having computer-executable instructions for performing steps comprising:
recognizing a history operand in source code;
finding at least one instance of the history operand in the source code in response to recognizing the history operand;
allocating storage; and
generating first object code associated with each instance, wherein the first object code, when executed, saves a data history associated with the history operand in the storage,

wherein the history operator is capable of being referenced directly by the source code.

27. (Original): The computer-readable medium of claim 26, further comprising:
 - recognizing a history operator in the source code; and
 - generating second object code that, when executed, performs the history operator on the data history.
28. (Original): The computer-readable medium of claim 26, wherein performing the history operator on the data history further comprises:
 - querying the data history based on contents of the data history.
29. (Original): The computer-readable medium of claim 26, wherein the history operand comprises a label associated with a source code statement, and wherein performing the history operator on the data history further comprises:
 - counting a number of times the source code statement associated with the label has been executed.
30. (Original): The computer-readable medium of claim 26, wherein performing the history operator on the data history is a function selected from a group consisting of:
 - summing the data history, averaging the data history, determining a maximum of the data history, determining a number of values in the data history, determining a first entry in the data history, determining a last entry in the data history, determining a subsequence of the data history, performing a reduction operation, and performing a statistical function.
31. (Currently Amended): A computer system comprising:
 - a processor;
 - memory coupled to the processor, wherein the memory contains a translator for translating source code into object code, wherein the translator comprises instructions, wherein the instructions when executed on the processor comprise:

recognizing a history operand in the source code, wherein the source code is contained in the memory;

in response to recognizing the history operand, finding at least one instance of the history operand in the source code;

allocating storage for a data history associated with the history operand;

generating first object code associated with each instance, wherein the first object code, when executed, saves the data history associated with the history operand in the storage; and

generating second object code that, when executed, performs a history operator on the data history,

wherein the history operator is capable of being referenced directly by the source code.

32. (Original): The computer system of claim 31, wherein the first object code further saves values assigned to a variable, wherein the variable is an instance of the history operand.

33. (Original): The computer system of claim 31, wherein the second object code, when executed, performs a function selected from a group consisting of:

summing the data history, averaging the data history, determining a maximum of the data history, determining a number of values in the data history, determining a first entry in the data history, determining a last entry in the data history, determining a subsequence of the data history, performing a reduction operation, and performing a statistical function.

34. (Original): The computer system of claim 31, wherein the history operand comprises a label associated with a source code statement, and wherein the second object code, when executed, further comprises:

counting a number of times the source code statement associated with the label has been executed.

35. Canceled

36. Canceled
37. Canceled
38. (Currently amended): A computerized method for interpreting source code, comprising:
recognizing a history operator and a history operand in the source code;
saving a data history associated with an instance of the history operand; and
performing the history operator on the data history,
wherein the history operator is capable of being referenced directly by the source code.
39. (Original): The method of claim 38, wherein saving the data history further comprises saving values assigned to a variable in the data history when the object code is executed.
40. (Original): The method of claim 38, wherein the history operand further comprises an expression of variables and wherein saving the data history further comprises saving a result of the expression in the data history.
41. (Original): The method of claim 38, wherein saving the data history further comprises allocating storage for the data history.
42. (Original): The method of claim 39, wherein the data history further comprises program locations where the assignments occurred and timestamps indicating when the assignment was made.
43. (Original): The method of claim 38, wherein performing the history operator on the data history further comprises:
querying the data history based on contents of the data history.
44. (Original): The method of claim 38, wherein the history operand comprises a function and the data history comprises values returned by the function.

45. (Original): The method of claim 44, wherein the data history further comprises program locations where the values were returned and timestamps indicating when the values were returned.
46. (Original): The method of claim 38, wherein the history operand comprises a label associated with a source code statement, and wherein performing the history operator on the data history further comprises:
counting a number of times the source code statement associated with the label was executed.
47. (Original): The method of claim 46, wherein the label is programmer-defined.
48. (Original): The method of claim 46, wherein the label comprises a programming language control construct.
49. (Original): The method of claim 38, wherein performing the history operator on the data history is a function selected from a group consisting of:
summing the data history, averaging the data history, determining a maximum of the data history, selecting an element of the data history, determining a minimum of the data history, determining a number of values in the data history, determining a first entry in the data history, determining a last entry in the data history, determining a subsequence of the data history, performing a reduction operation, and performing a statistical function.
50. (Original): The method of claim 38, wherein:
the history operand comprises a programming language keyword representing a loop; and
the history operator comprises an iteration count of the loop.
51. (Original): The method of claim 38, wherein the saving of the data history further comprises:

saving the data history in an array, wherein each element of the array comprises a value associated with the history operand at a particular time.

52. (Original): The method of claim 38, wherein saving the data history further comprises:

saving the data history in a linked list.

53. (Original): The method of claim 38, wherein saving the data history further comprises:

saving the data history in a file.

54. (Original): The method of claim 38, wherein performing the history operator on the data history further comprises:

resetting the data history to null.

55. (Currently Amended): A computer-readable medium having computer-executable instructions for performing steps comprising:

recognizing a history operand in source code, the history operand representing a sequence of data associated with the history of an operand instance;

finding at least one instance of the history operand in the source code in response to recognizing the history operand; and

saving a data history associated with each instance of the history operand,
wherein the history operand is capable of being referenced directly by the source code.

56. (Previously Presented): The computer-readable medium of claim 55, further comprising:

recognizing a history operator in the source code, the history operator representing a function that object code will perform on the data history associated with the history operand; and

performing the history operator on the data history.

57. (Original): The computer-readable medium of claim 55, wherein performing the history operator on the data history further comprises:

querying the data history based on contents of the data history.

58. (Original): The computer-readable medium of claim 55, wherein the history operand comprises a label associated with a source code statement, and wherein performing the history operator on the data history further comprises:

counting a number of times the source code statement associated with the label has been executed.